

Mecanismo de aprendizaje automático: red neuronal perceptrón utilizando Matlab

Norma Verónica Ramírez Pérez

Instituto Tecnológico de Celaya
norma.ramirez@itcelaya.edu.mx

Martín Laguna Estrada

Instituto Tecnológico de Celaya
martin.laguna@itcelaya.edu.mx

Carlos Alberto Ruiz Ayala

Instituto Tecnológico de Celaya
carlosalb_14@hotmail.com

RESUMEN

Las redes neuronales artificiales(RNA), han sido utilizadas en diferentes ámbitos y disciplinas, una de sus aplicaciones más comunes es su uso como herramienta de análisis de información. En este trabajo se hace una breve introducción sobre las particularidades de las RNA, específicamente las basadas en el modelo perceptrón simple, y se exponen los elementos que la integran y se vincula a una aplicación sobre clasificación de la base de datos Iris con la finalidad de mostrar su funcionamiento.

PALABRAS CLAVE: perceptrón, redes neuronales artificiales, Matlab, mecanismos de clasificación de datos.

1. INTRODUCCIÓN

El creciente desarrollo e implementación de nuevas técnicas computacionales han marcado desde sus inicios un cambio en el campo de la informática. Una de las áreas favorecidas con este desarrollo ha sido el de la inteligencia artificial y en particular, en el

campo de la lógica borrosa (aprendizaje inductivo), algoritmos genéticos y redes neuronales. Sus aplicaciones hasta el día de hoy van desde la industria de la producción, hasta la industria de juegos y a infinidad de aplicaciones en aparatos de uso cotidiano, y es por esta razón la importancia que toma el estudio de temas relacionados con la inteligencia artificial.

A través de los años, prácticamente desde los años 50, se fueron desarrollando importantes investigaciones en el campo de la inteligencia artificial, sobre todo con aquellas que tenían como objeto principal las redes neuronales artificiales (RNA). El autor Rosenblatt[1], en sus nuevas publicaciones acerca del perceptrón, abrió nuevas expectativas sobre este tema. No obstante, las teorías desarrolladas por Minski y Papert[2], hacen una crítica al modelo neuronal y tratan de plasmar lo estériles que eran las investigaciones en este tema. En la década de los 70's, retoman fuerza estas teorías y se desarrolla un interés creciente para su aplicación en diferentes campos.

El objetivo del presente trabajo es utilizar la base teórica de las redes neuronales artificiales, como una herramienta práctica que permita realizar análisis exploratorios de datos o minería de datos, vinculados con la clasificación utilizando el algoritmo perceptrón simple.

2. METODOLOGÍA DE ANÁLISIS DE LAS RNA ESTRUCTURADA EN BASE AL PERCEPTRÓN.

2.1. Redes neuronales artificiales

Para los autores Freman y Skapura [3], una red neuronal es un sistema de

procesadores paralelos conectados entre si en forma de grafo dirigido. Su esquema de cada elemento de procesamiento denominado 'neuronas' de la red, se presenta como un nodo. Dichas conexiones establecen una estructura jerárquica tratando de emular la fisiología del cerebro humano y buscar nuevos modelos de procesamiento para solucionar problemas concretos del mundo real. Por la versatilidad de las RNA y debido a su comportamiento para aprender, reconocer y aplicar relaciones entre objetos y tramas de objetos propios de la vida cotidiana, son utilizadas como herramienta que puede llegar a resolver problemas difíciles, tanto de clasificación como de reconocimiento.

Dicha posibilidad de resolución de problemas difíciles, se debe a los principios de las redes neuronales, que cuentan con las siguientes características que cita Hilera y Martínez [4]:

- **Aprendizaje adaptativo.** Es una característica importante de las RNA, debido al comportamiento de entrenamiento con una serie de ejemplos ilustrativos, de esta forma, no es necesario elaborar un modelo a priori, ni el establecimiento de funciones probabilísticas. Una RNA es adaptativa, dado que puede modificarse en repetidas ocasiones con el fin de obtener nuevas condiciones de trabajo.
- **Autoorganización.** Mientras que el aprendizaje es un proceso donde se modifica información interna de la red neuronal artificial, la autoorganización consiste en la modificación de la red completa con el fin de llevar a cabo un objetivo específico. Autoorganización significa generalización, de esta forma una red puede responder a datos o situaciones que no ha experimentado antes, pero

que puede inferir sobre la base de su entrenamiento. Esta característica es muy útil sobre todo cuando la información de entrada es poco clara o se encuentra incompleta.

- **Tolerancia a fallos.** La razón por la que las redes neuronales son tolerantes a los fallos, es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento. La mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos almacenan cada pieza de información en un espacio único, localizado y direccionable. En cambio, las redes neuronales almacenan información no localizada. De ahí que, la mayoría de las interconexiones entre los nodos de la red tendrán sus valores en función de los estímulos recibidos, y se generará un patrón de salida que represente la información almacenada.
- **Operación en tiempo real.** Una de las mayores prioridades, casi en la totalidad de las áreas de aplicación, es la necesidad de realizar procesos con datos de forma muy rápida. Las redes neuronales se adaptan bien a esto debido a su implementación paralela. Para que la mayoría de las redes puedan operar en un entorno de tiempo real, la necesidad de cambio en los pesos de las conexiones o entrenamiento es mínimo.

Para profundizar un poco en el tema de las RNA, se puede mencionar que el término neurona no es de extrañar que se piense que se está uno refiriendo al cerebro humano, y esto es debido a que las RNA están inspiradas biológicamente. Se dice que el hombre posee cerca de diez millones de neuronas masivamente interconectadas,

entendiendo que la neurona es una célula que está especializada para propagar una señal electroquímica.

La estructura ramificada de las neuronas en cuanto a las entradas (dendritas en el cerebro humano) y la estructura ramificada de salida (axones en el cerebro humano), se conectan entre sí, es decir, los axones de una célula se concretan con las dendritas de otra por vía de la sinapsis, de ahí que el principio de las RNA sea imitar el funcionamiento cerebral.

2.2. Estructura de las RNA

Las RNA están formadas por una gran cantidad de neuronas que suelen denominarse nodos o unidades procesadoras. Un nodo o neurona cuenta con una cantidad variable de entradas (X_1, X_2, \dots, X_m) y a su vez dispone de una sola salida (Y_i) , la cual transmite la información al exterior o hacia otras neuronas. Cada salida o señal de salida tiene asociada un peso que se calculará en función de las entradas, por lo que cada una de ellas es afectada por un determinado peso definido por w_i , con $i = 1, 2, \dots, n, n+1$, que modifica las entradas antes de que sean sumadas y suministradas al elemento de umbral. En este sentido, los pesos son similares a las sinapsis en el sistema neuronal humano. La función que transforma la salida correspondiente a la suma en la salida final, se denomina *función de activación* y toma el valor de 1 si su argumento es verdadero y -1 si es falso, como podemos ver en la figura 1. Pajares, Santos [5].

El proceso de aprendizaje consiste en hallar los pesos que codifican los conocimientos. Una regla de aprendizaje hace variar el valor de los pesos de una red hasta que estos

adoptan un valor constante, cuando esto ocurre se dice que la red ya "ha aprendido".

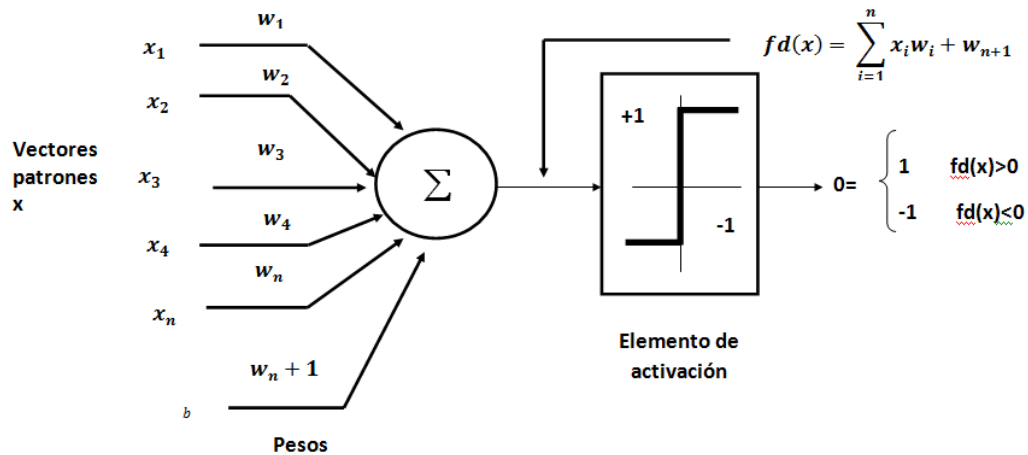


Figura 1. Modelo de perceptrón.

Cuando se conectan varias neuronas, se consigue una red que puede presentar diversas variaciones en las tipologías, las cuales son clasificadas según los criterios que se exponen a continuación:

- Número de niveles o capas.
- Número de neuronas por nivel.
- Formas de conexión.

De acuerdo al problema que se vaya a solucionar, el diseño de una u otra tipología variará. Si tomamos como ejemplo la elaboración de un programa de filtro digital en una computadora, se deberá emplear un algoritmo que permita que todas las capas estén uniformemente interconectas, es decir, que todos los nodos de una capa estén conectados con los nodos de otra capa. Para hacer referencia a este tipo de red, en la figura 2 se muestra la arquitectura de una RNA clásica con variables de entrada, dos capas intermedias y una capa de salida, todas ellas conectadas entre sí.

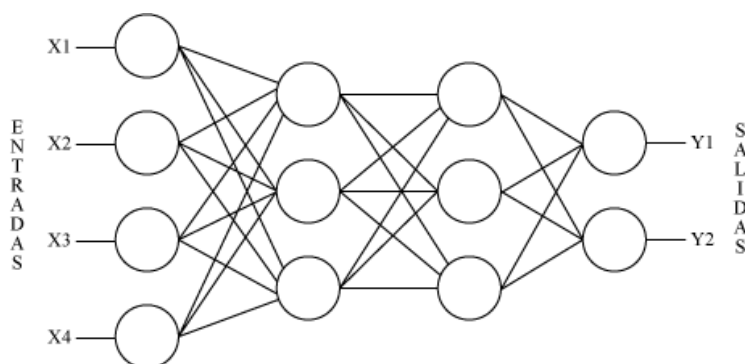


Figura 2. Esquema de una RNA de dos capas de neuronas intermedias.

Existe una gran variedad de modelos de RNA que dependen del objetivo para los cuales fueron diseñados y del problema práctico a solucionar. En la actualidad, el éxito de las RNA, depende muchas veces del usuario, para lograr alcanzar aplicaciones que sean de gran utilidad y porque no, hasta valor comercial. Sin embargo, hay algunos modelos más populares que otros, uno de estos modelos es referenciado por el autor Pao YH 1995[6], quien menciona que el modelo Adeline/Madaline fue diseñado por Bernard Widrow en 1956 y es utilizado en el diseño y realización de filtros, así como en la eliminación de ruido en señales portadoras de información, como es el caso de los módems. El backpropagation es utilizado en el proyecto de la máquina de escribir neural fonética, y el modelo de los mapas auto-organizados (Self-Organizing Map, SOM), mejor conocido como modelo de Kohonen por ser éste su creador [7], y el perceptrón simple, utilizado en la clasificación de clases. Este último es el que se utilizará para hacer una aplicación del funcionamiento de este tipo de red, haciendo uso de la base de datos iris de la UC Irvine Machine Learning Repository[8], en la cual se encuentra una gran variedad de datos que son utilizados para la demostración del funcionamiento de los algoritmos de aprendizaje automático.

3. APLICACIÓN DE RED NEURONAL PERCEPTRÓN UTILIZANDO MATLAB

El propósito de este trabajo es simular una red neuronal, así como conocer las funciones con las cuales se implementa en Matlab.

A continuación se muestra un algoritmo de pasos para el desarrollo de la aplicación de red neuronal perceptrón utilizando Matlab.

Desarrollo:

1) Datos de entrada: X_1, X_2, X_3 .

Las clases se representan de la forma:

Iris-setosa = 0

Iris-versicolor = 1

La tabla 1 muestra un sector de los datos a utilizar en esta aplicación, los cuales representan las variables a clasificar así como las clases correspondientes.

Tabla 1. Muestra parcial de la base de datos iris.

X1	X2	X3	clase
5.1	3.5	1.4	0
4.9	3.0	1.4	0
4.7	3.2	1.3	0
4.6	3.1	1.5	0
5.0	3.6	1.4	0
...
...
5.6	2.7	4.2	1

5.7	3.0	4.2	1
5.7	2.9	4.2	1
6.2	2.9	4.3	1
5.1	2.5	3.0	1
5.7	2.8	4.1	1

Para la finalidad de esta aplicación, se consideró conveniente almacenar todos los datos de las entradas y las clases en un archivo de Excel formando un conjunto de celdas de tamaño 4 x 100.

- 2) Crear una matriz de entradas en MATLAB importando los valores desde el archivo de Excel, aprovechando la interacción que se tiene entre estos programas.
- 3) Generar el vector de las clases (salidas deseadas).

$$\text{entrada}=[X_1, X_2, X_3], \text{ salida}=[0, 1]$$

- 4) Graficar los datos de entrada y salida `plotpv(entrada, salida)`

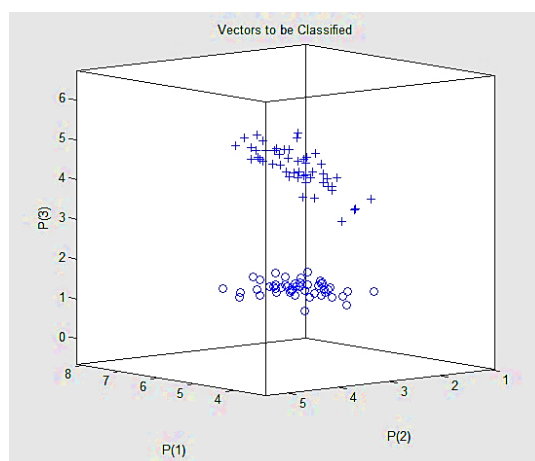


Figura 3. Clasificación de vectores

- 5) Crear una red neuronal tipo perceptron.

```
net = newp(minmax(entrada),1);
```

- 6) Una vez creada la red, se tiene la opción de ir asignando valores arbitrarios de entrada al vector de pesos y al parámetro bias para ir verificando la clasificación de los datos. Para el vector de pesos, se asigna un renglón con las columnas necesarias para representar las entradas (X_1, X_2, X_3) . El vector pesos y el parámetro bias son dos asignaciones que se pueden modificar internamente en una RNA, a los cuales se asignaron valores de 1,1,1 a cada uno de los pesos y se le asignó un valor de -10.5 en el bias.

- 7) Asignar valores a los vectores de pesos y bias para facilitar la interpretación del siguiente paso: *plotpc(pesos,bias)* y así crear una gráfica lineal de separación que genera el perceptrón como primera instancia según los valores asignados a los *pesos* y al parámetro *bias*. Dicha interpretación se puede observar en la figura 4, pero como aún no se ha entrenado la red, la línea de discriminación no separará las clases. Como nota adicional, en esta aplicación se debe tener cuidado con la longitud del vector de pesos que no exceda de tres valores para poder graficarlo, de lo contrario será imposible.

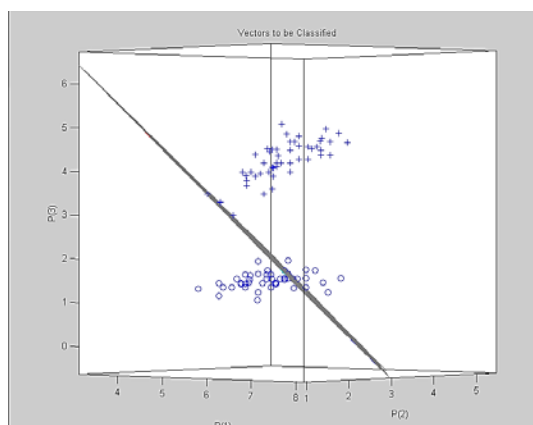


Figura 4. El plano agregado no muestra una clasificación correcta de los datos de entrada, por lo tanto es errónea.

- 8) El siguiente paso, es el entrenar la red para que mejore su aprendizaje y pueda dar una clasificación con el mínimo de error o en el mejor de los casos, sin error.

Lo anterior se puede realizar con la siguiente instrucción

```
net = train(net, entrada, salida)
```

4. RESULTADOS

En la parte final del análisis, se verifica visualmente que tan bien se realizó la clasificación, esto se puede comprobar haciendo un nuevo ploteo, con *plotpv* y *plotpc*, para la generación de nueva gráfica que permita ilustrar los datos ya entrenados de la red. En la figura 5 se puede observar que la gráfica a) presenta la clasificación sin entrenamiento de los datos, mientras que en la gráfica b), es posible observar la línea de discriminación después de entrenar la red en donde se ve claramente la clasificación de los datos, con lo cual es posible observar la eficiencia del método utilizado para la clasificación de datos mediante el perceptrón.

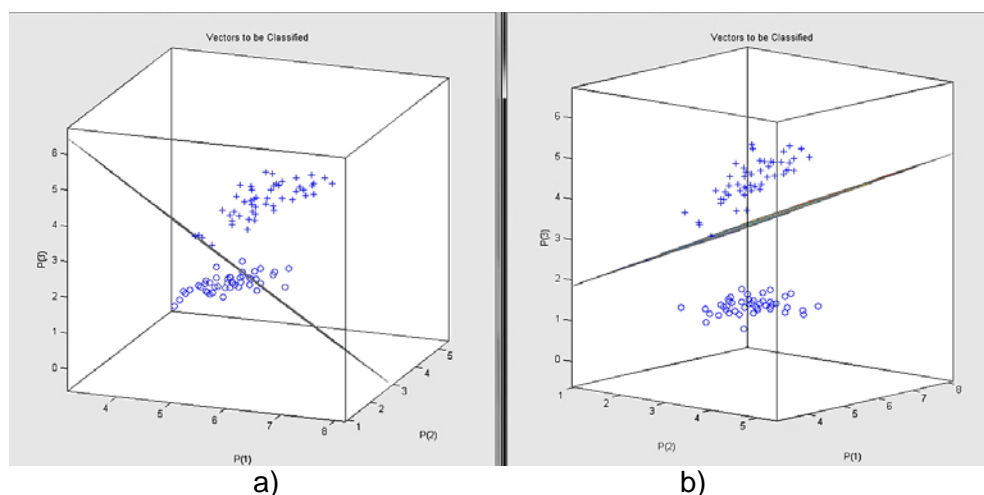


Figura 5. Plano de distribución con la clasificación correcta de clases.

5. CONCLUSIONES

La idea principal de esta contribución ha sido mostrar a través de una aplicación, el uso de bases de datos a través de máquinas de aprendizaje automático, las cuales son un área común en la inteligencia artificial. Para este fin se diseñó una aplicación con la herramienta toolbox que viene integrada en Matlab, haciendo una separación de clases del archivo de datos iris, la cual es una de las bases de datos más conocida en la literatura de reconocimiento de patrones y es muy manejada en el ámbito académico para las pruebas de diferentes algoritmos de clasificación. Para ejemplificar la RNA, sólo se tomaron tres de los cuatro atributos que originalmente trae la base de datos por conveniencia para hacer la demostración de este método. Los datos lograron ser clasificados por la RNA de tipo percentrón simple, mediante el cual se pudo probar su clasificación con éxito. De igual manera se comprobó la eficiencia de este tipo de algoritmos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65(3):386-408.
- [2] Minsky, M.; Papert, S. *Perceptrons*. Massachusetts: Cambridge University; 1969; 226 p.
- [3] Freeman, J.A.; Skapura, DM. *Redes Neuronales. Algoritmos, aplicaciones y técnicas de propagación*. México: Addison-Wesley. 1993; 306 p.

- [4] Hilera González, J.; Martínez Hernández, V. Redes neuronales artificiales: fundamentos, modelos y aplicaciones. Madrid, RA-MA, 1995. 389 p.
- [5] Pajares, M.; Santos, M. Inteligencia artificial e Ingeniería del conocimiento, Madrid, RA-MA 2006, 235 p.
- [6] Pao Y.H., Neural Net Computing for pattern Recognition . In Chen , L.F. Pau and P.S.P. Wang (Eds) Handbook of Pattern Recognition and Computer Vision, 1995; 125-162 p.
- [7] Kohonen, T. Self-Organizing-Maps. Springer, Berlin, Second edition. 1997, 362 p.
- [8] Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.